WHAT IS CLAIMED IS:

1. An authentication method providing a data signing function that determines an authentication tag for use in conjunction with transfer of data using a communication channel or with data storage on storage media, comprising the steps of:

partitioning said data into a plurality of data blocks;

for each of said data blocks, performing a randomization function over said data block to create an input block of the same size as that of said data block, said input block not including a block identifier;

applying a pseudo-random function to each said input block to create a plurality of enciphered blocks; and

combining said plurality of enciphered blocks to create an authentication tag.

2. The method of claim 1, wherein the pseudo-random function is a standard block cipher.

3. The method of claim 1, wherein each of said data blocks is ℓ bits in length.

4. The method of claim 3, comprising the step of creating a random vector block of ℓ bits in length.

5. The method of claim 4, wherein the step of performing a randomization function over said plurality of data blocks includes performing the randomization function over the random vector block.

6. An authentication method providing a data signing function that determines an authentication tag, comprising the steps of:

receiving an input plaintext string comprising the data to be signed and padding it as necessary such that its length is a multiple of ℓ bits;

partitioning the padded input plaintext string into a plurality of equal-size plaintext blocks of ℓ bits in length;

creating a random vector of ℓ bits in length;

performing a randomization function over said plurality of plaintext blocks and the random vector block to create a plurality of input blocks each of ℓ bits in length;

applying a block cipher using a secret key over each of said input blocks to create a plurality of enciphered blocks each of L bits in length; and

performing a combination operation over said plurality of enciphered blocks to create an authentication tag.

7. The method as defined in claim 6, comprising the steps of:

wherein said performing a randomization function step comprises combining each of said plaintext blocks and the random vector block with a different corresponding element of a sequence of unpredictable elements to create a plurality of input blocks.

8. The method as defined in claim 7, comprising the step of generating said random vector block from a random number generated on a per-message basis.

9. The method as defined in claim 7, further comprising the step of appending the created random vector block after a last block of the set of equal-sized blocks comprising the padded plaintext string.

10. The method as defined in claim 7, wherein the input blocks from the randomization step comprise n + 1 blocks each of ℓ-bit length,

where n is the total number of blocks in said set of equal-sized blocks of the padded input plaintext string.

11. The method as defined in claim 7, further comprising the step of generating each of a plurality of the unpredictable elements of said sequence of unpredictable elements by combining a different element index i of each of the unpredictable elements and a random initial vector.

12. The method as defined in claim 11, comprising the step of generating said random initial vector from a random number generated on a per-message basis.

13. The method as defined in claim 8, further comprising the steps of:

wherein said sequence of the unpredictable elements is generated by combining a different element index i of each of the unpredictable elements and a random initial vector; and

wherein said random initial vector is generated from said random number.

14. The method as defined in claim 8, further comprising the steps of:

enciphering a random number using the block cipher using the secret key to generate a random initial vector;

using this random initial vector to generate the elements of the sequence of unpredictable elements.

15. The method of claim 7, wherein said random vector is generated by enciphering a random number of ℓ bits in length, said enciphering using said block cipher using a secret second key.

16. The method as defined in claim 8, wherein said random vector is generated by enciphering a variant of said random number of ℓ bits in length, said enciphering using said block cipher using said secret key.

17. The method as defined in claim 16, wherein said variant of said random number is obtained by adding a non-zero constant to said random number.

18. The method of claim 8, further comprising the steps of:

wherein the random number is provided by a random number generator; and

outputting the random number as an output block of the authentication scheme.

19. The method as defined in claim 7, further comprising:

generating said random initial vector by enciphering a count of a counter initialized to a constant, said enciphering being performed with the block cipher using the secret key;

generating said random vector block from said count of a counter; and

incrementing said counter by one on every message signing.

20. The method as defined in claim 19, wherein said random vector block is generated by enciphering said count of a counter using a second secret key.

21. The method as defined in claim 19, wherein said random vector is generated by enciphering a variant of said count of a counter, said enciphering using said block cipher using said secret key.

22. The method as defined in claim 21, wherein said variant of said of said count of a counter is obtained by adding a non-zero constant to said count of counter.

23. The method as defined in claim 19, wherein said counter is initialized to a constant whose value is the ℓ-bit representation of negative one.

24. The method as defined in claim 19, further comprising:

outputting said counter value as an output block of the authentication scheme.

25. The method as defined in claim 7, further comprising the steps of:

wherein the random vector is generated from a shared, per-key, random initialization vector and the count of a counter;

incrementing said counter by one on every message signing, wherein said counter is initialized to a constant whose value is the ℓ-bit representation of negative one; and

outputting said counter value as an output block of the authentication scheme.

26. The method as defined in claim 6, wherein said combination operation comprises a bit-wise exclusive-or operation.

27. The method as defined in claim 6, wherein said combination operation comprises an addition modulo $2^L - 1$.

28. The method as defined in claim 6, wherein said combination operation comprises a subtraction modulo $2^L - 1$.

29. The method as defined in claim 7, wherein said combining step to create a plurality of input blocks comprises an addition modulo 2^{ℓ} operation.

30. The method as defined in claim 7, wherein said combining step to create a plurality of input blocks comprises a bit-wise exclusive-or operation.

31. The method as defined in claim 7, wherein said combining step to create a plurality of input blocks comprises a subtraction modulo 2^{ℓ} operation.

32. The method as defined in claim 7, further comprising:

generating a random initial vector from a random number of ℓ-bit length; and

generating each element in said sequence of unpredictable elements by modular 2^{ℓ} multiplication of a different unique element identifier (i) for each element in the sequence of unpredictable elements and said random initial vector.

33. The method as defined in claim 7, further comprising:

generating a random initial vector from a random number of ℓ-bit length; and

generating each element in said sequence of unpredictable elements from the previous element by modular 2^{ℓ} addition of said random initial vector to the previous element, with a first element of said sequence being said random initial vector itself.

34. The method of claim 6, wherein said performing a randomization function over said plurality of plaintext blocks and the random vector block is done concurrently for each plaintext block and the random vector block.

35. The method of claim 6 wherein the plurality of input blocks resulting from performing a randomization function over said plurality of plaintext blocks and the random vector block are concurrently presented to a plurality of block ciphers using a secret key.

36. An authentication method providing a data signing function that determines an authentication tag, comprising the steps of:

receiving an input plaintext string comprising the data to be signed and padding it as necessary such that its length is a multiple of ℓ bits;

partitioning the padded input plaintext string into a plurality of n equal-size plaintext blocks of ℓ bits in length;

performing a randomization function over said plurality of n plaintext blocks to create a plurality of input blocks each of ℓ bits in length;

applying a block cipher using a secret key over each of the said input blocks to create a plurality of enciphered blocks each of L bits in length; and

performing a combination operation over said plurality of enciphered blocks to create an authentication tag.

37. The method of claim 36, wherein said step of performing a randomization function over said plurality of n plaintext blocks comprises combining each of said plurality of plaintext blocks with a different corresponding element of a plurality of n unpredictable elements to create a plurality of input blocks.

38. The method of claim 36, wherein each of the said plurality of n unpredictable elements is obtained by applying an operation to a different per-message unpredictable element and each of a plurality of internal unpredictable elements.

39. The method of claim 38, further comprising the steps of:

wherein said per-message unpredictable element is obtained from an ℓ-bit counter and a secret, first random initial vector shared between sender and receiver;and

wherein each of said plurality of internal unpredictable elements is obtained from an ℓ-bit element index and a secret, second random initial vector shared between sender and receiver.

40. The method of claim 38, wherein said operation applied to a different per-message unpredictable element and each of a plurality of internal unpredictable elements comprises an addition modulo 2^{ℓ} operation.

41. The method of claim 38, wherein said operation applied to a different per-message unpredictable element and each of a plurality of internal unpredictable elements comprises a subtraction modulo 2^{ℓ} operation.

42. The method of claim 38, wherein said operation applied to a different per-message unpredictable element and each of a plurality of internal unpredictable elements comprises a bit-wise exclusive-or operation.

43. The method of claim 39, further comprising the steps of:

wherein said per-message unpredictable element is obtained by multiplication modulo 2^{ℓ} of said secret, first random initial vector with a different value of the counter; and

wherein each of said plurality of internal unpredictable elements is obtained by multiplication modulo 2^{ℓ} of said secret, second random initial vector with a different value of the index.

44. The method of claim 39, further comprising the steps of:

wherein said per-message unpredictable element is obtained from the previous per-message unpredictable element by modular 2^{ℓ} addition of said first random initial vector to the previous per-message unpredictable element, with a first per-message unpredictable element being said first random initial vector itself; and

wherein each of said plurality of internal unpredictable elements is obtained from the previous internal unpredictable element by modular 2^{ℓ} addition of said second random initial vector to the previous internal unpredictable element, with a first internal unpredictable element being said second random initial vector itself.

45. The method of claim 37, wherein said combining step to create a plurality of input blocks comprises an addition modulo 2^{ℓ} operation.

46. The method of claim 37, wherein said combining step to create a plurality of input blocks comprises a subtraction modulo 2^{ℓ} operation.

47. The method of claim 37, wherein said combining step to create a plurality of input blocks comprises a bit-wise exclusive-or operation.

48. The method of claim 38, comprising:

generating said counter anew for every new key;

initializing generated counter to a constant value; and

for each message being signed using key, incrementing said counter by the one; and

outputting said counter as an output block of the authentication scheme.

49. The method as defined in claim 36, wherein said combination operation comprises a bit-wise exclusive-or operation.

50. The method as defined in claim 36, wherein said combination operation comprises an addition modulo $2^L - 1$.

51. The method as defined in claim 36, wherein said combination operation comprises a subtraction modulo $2^L - 1$.

52. A verification method for the authentication method, which provides data integrity, comprising the steps of:

presenting a string including a plaintext string and an input authentication tag for verification;

partitioning said plaintext string into a plurality of n plaintext blocks comprising ℓ bits each;

performing the same randomization function as that used at a signing method for determining an authentication tag over said plurality of plaintext blocks to create a plurality of input blocks each of ℓ bits in length;

applying a block cipher using a secret key over each of the said input blocks to create a plurality of enciphered blocks each of L bits in length;

performing the same combination operation as that used at a signing method for determining an authentication tag over said plurality of enciphered blocks to compute an authentication tag;

verifying integrity of the plaintext blocks by comparing the input authentication tag and the computed authentication tag.

53. The method as defined in claim 52, further comprising the steps of:

creating a secret random vector block of ℓ bits in length;

performing the same randomization function as that used at a signing method for determining an authentication tag over said plurality of plaintext blocks and the secret random vector block to create a plurality of input blocks each of ℓ bits in length;

wherein performing said randomization function further comprises:

deriving a random initial vector from said string presented for decryption;

generating a sequence of unpredictable elements each of ℓ-bit length from said random initial vector in the same manner as used at signing method; and

selecting n plaintext blocks from said string in the same order as that used at the signing method, and combining said selected plaintext blocks and the random vector with a different corresponding element of said sequence of unpredictable elements to obtain a plurality of input blocks, in the same manner as that used at the signing method.

54. The method as defined in claim 52, wherein performing said randomization function further comprises:

using a secret, random initial vector shared between sender and receiver;

generating a sequence of unpredictable elements each of ℓ-bit length from said secret, random initial vector in the same manner as used at signing method; and

selecting n plaintext blocks from said string in the same order as that used at the signing method, and combining said selected plaintext blocks with a different corresponding element of said sequence of unpredictable elements to obtain a plurality of input blocks, in the same manner as that used at the signing method.

55. The method of claim 52, further comprising:

selecting one block of the from said string presented for authentication, which block contains a random number; and

enciphering the selected block to obtain the random initial vector using the block cipher using a first secret key.

56. The method of claim 52, further comprising:

for the signing method generating a random initial vector by enciphering a count of a counter initialized to a constant, said enciphering being performed with the block cipher using a secret key; and

incrementing said counter by one on every message signing; and

further comprising for authentication of the partitioned plaintext string the steps of:

selecting a counter block representing the count of the counter from said string presented at verification; and

enciphering said selected counter block to obtain a random initial vector.

57. The method as defined in claim 56, wherein the enciphering step comprises performing said enciphering using the block cipher using the secret key.

58. An authentication method providing a data signing function that updates an authentication tag incrementally, comprising the steps of:

receiving an input plaintext string comprising the data to be signed and padding it as necessary such that its length is a multiple of ℓ bits;

partitioning the padded input plaintext string into a plurality of equal-size plaintext blocks of ℓ bits in length;

performing a randomization function over said plurality of plaintext blocks to create a plurality of input blocks each of ℓ bits in length;

applying a block cipher using a secret key over each of the said input blocks to create a plurality of enciphered blocks each of L bits in length;

performing a combination operation over said plurality of enciphered blocks to create an authentication tag, said combination operation having an inverse; and

further comprising the steps of:

receiving an input plaintext string including a plaintext string and an input authentication tag;

partitioning said plaintext string into a plurality of n plaintext blocks comprising ℓ bits each;

receiving a new ℓ-bit input plaintext block to replace an ℓ-bit plaintext block of said input plaintext string at index i;

performing the same randomization function as that used at a signing method, using index i, on said new input plaintext block to create a first input block and performing the same randomization function as that used at a signing method, using index i, on said plaintext block at index i to create a second input block, each of the said created input blocks having ℓ bits in length;

applying a block cipher using a secret key to the first input block and the second input block to create a first enciphered block and a second enciphered block, each of L bits in length;

performing the inverse of said combination operation used at a signing method for determining an authentication tag to the input authentication tag and said second enciphered block;

performing the said combination operation used at a signing method for determining an authentication tag to first enciphered block and the result of performing the inverse of said combination operation; and

outputting the result of performing said combination operation to the first enciphered block and the result of performing the inverse of said combination operation as the authentication tag.

59. The method of claim 58 comprising the steps of:

receiving a plurality of new ℓ-bit input plaintext blocks to replace a plurality of ℓ-bit plaintext blocks of said input plaintext string at index i; and

providing a data signing function that determines an authentication tag incrementally for each of the said plurality of new ℓ-bit input plaintext blocks.

60. An authentication method providing a data signing function that determines an authentication tag, comprising the steps of:

receiving an input plaintext string comprising the data to be signed and padding it as necessary such that its length is a multiple of ℓ bits;

partitioning the padded input plaintext string into a plurality of equal-size plaintext blocks of ℓ bits in length;

performing a randomization function over each of said plurality of plaintext blocks using a different index for each plaintext block to create a plurality of input blocks each of ℓ bits in length;

applying a block cipher using a secret key over each of the said input blocks to create a plurality of enciphered blocks each of L bits in length;

performing a combination operation over said plurality of enciphered blocks to create an authentication tag; and

further providing an out-of-order verification function for the authentication method comprising the steps of:

receiving an input authentication tag for verification and a plurality of n plaintext blocks comprising ℓ bits each, each plaintext block being accompanied by a different index;

performing a randomization function over each of said plurality of plaintext blocks using said index for each plaintext block to create a plurality of input blocks each of ℓ bits in length;

applying a block cipher using a secret key over each of the said input blocks to create a plurality of enciphered blocks each of L bits in length;

performing the same combination operation as that used at a signing method for determining an authentication tag over said plurality of enciphered blocks to compute an authentication tag;

verifying integrity of the plaintext blocks by comparing the input authentication tag and the computed authentication tag.

61. An authentication system for providing a data signing function that determines an authentication tag for use in conjunction with transfer of data using a communication channel or with data storage on storage media, comprising:

a partitioner for partitioning said data into a plurality of data blocks;

a randomization component which, for each of said data blocks, performs a randomization function over said data block to create an input block of the same size as that of said data block, said input block not including a block identifier;

a pseudo-random encipher component for applying a pseudo-random function to each said input block to create a plurality of enciphered blocks; and

a combining component for combining said plurality of enciphered blocks to create an authentication tag.

62. The system of claim 61, wherein the pseudo-random encipher component applies a pseudo-random function that is a standard block cipher.

63. An authentication system for providing a data signing function that determines an authentication tag, comprising:

a partitioner for partitioning an input plaintext string into a plurality of equal-size plaintext blocks of ℓ bits in length;

a first component for creating a random vector of ℓ bits in length;

a second component for performing a randomization function over said plurality of plaintext blocks and the random vector block to create a plurality of input blocks each of ℓ bits in length;

a block cipher component for applying a block cipher using a secret key over each of said input blocks to create a plurality of enciphered blocks each of L bits in length; and

a combining component for performing a combination operation over said plurality of enciphered blocks to create an authentication tag.

64. An authentication system for providing a data signing function that determines an authentication tag, comprising:

a partitioning component for partitioning a padded input plaintext string into a plurality of n equal-size plaintext blocks of ℓ bits in length;

a first component for performing a randomization function over said plurality of n plaintext blocks to create a plurality of input blocks each of ℓ bits in length;

a second component for applying a block cipher using a secret key over each of the said input blocks to create a plurality of enciphered blocks each of L bits in length; and

a combining component for performing a combination operation over said plurality of enciphered blocks to create an authentication tag.

65. A verification system for an authentication method, which provides data integrity, comprising:

a receiver for receiving a string including a plaintext string and an input authentication tag for verification;

a partitioner component for partitioning said plaintext string into a plurality of n plaintext blocks comprising ℓ bits each;

a first component for performing the same randomization function as that used at a signing method for determining an authentication tag over said plurality of plaintext blocks to create a plurality of input blocks each of ℓ bits in length;

a second component for applying a block cipher using a secret key over each of the said input blocks to create a plurality of enciphered blocks each of L bits in length;

a combining component for performing the same combination operation as that used at a signing method for determining an authentication tag over said plurality of enciphered blocks to compute an authentication tag; and

a comparator for verifying integrity of the plaintext blocks by comparing the input authentication tag and the computed authentication tag.

66. The system as defined in claim 65, further comprising:

a third component for creating a secret random vector block of ℓ bits in length;

wherein the first component performs the same randomization function as that used at the signing method over said plurality of plaintext blocks and the secret random vector block to create a plurality of input blocks each of ℓ bits in length;

wherein the first component performing said randomization function further comprises:

a component for deriving a random initial vector from said string presented for decryption;

a component for generating a sequence of unpredictable elements each of ℓ-bit length from said random initial vector in the same manner as used at signing method; and

a component for selecting n plaintext blocks from said string in the same order as that used at the signing method, and combining said selected plaintext blocks and the random vector with a different corresponding element of said sequence of unpredictable elements to obtain a plurality of input blocks, in the same manner as that used at the signing method.

67. The system as defined in claim 65, wherein the first component for performing said randomization function further comprises:

a component for using a secret, random initial vector shared between sender and receiver;

a component for generating a sequence of unpredictable elements each of ℓ-bit length from said secret random initial vector in the same manner as used at signing method; and

a component for selecting n plaintext blocks from said string in the same order as that used at the signing method, and combining said selected plaintext blocks with a different corresponding element of said

sequence of unpredictable elements to obtain a plurality of input blocks, in the same manner as that used at the signing method.

68. An authentication system for providing a data signing function that updates an authentication tag incrementally, comprising:

a partitioner for partitioning an input plaintext string into a plurality of equal-size plaintext blocks of ℓ bits in length;

a first component for performing a randomization function over said plurality of plaintext blocks to create a plurality of input blocks each of ℓ bits in length;

a block cipher component for applying a block cipher using a secret key over each of the said input blocks to create a plurality of enciphered blocks each of L bits in length;

a combining component for performing a combination operation over said plurality of enciphered blocks to create an authentication tag, said combination operation having an inverse; and

further comprising:

a receiver for receiving an input plaintext string including a plaintext string and an input authentication tag;

a partitioner component for partitioning said plaintext string into a plurality of n plaintext blocks comprising ℓ bits each;

a second receiver for receiving a new ℓ-bit input plaintext block to replace an ℓ-bit plaintext block of said input plaintext string at index i;

a component for performing the same randomization function as that used at a signing method, using index i, on said new input plaintext block to create a first input block and performing the same randomization function as that used at a signing method, using index i, on said plaintext block at index i to create a second input block, each of the said created input blocks having ℓ bits in length;

a third component for applying a block cipher using a secret key to the first input block and the second input block to create a first enciphered block and a second enciphered block, each of L bits in length;

a fourth component for performing the inverse of said combination operation used at a signing method for determining an authentication tag to the input authentication tag and said second enciphered block;

a fifth component for performing the said combination operation used at a signing method for determining an authentication tag to first enciphered block and the result of performing the inverse of said combination operation; and

a sixth component for outputting the result of performing said combination operation to the first enciphered block and the result of performing the inverse of said combination operation as the authentication tag.

69. The system of claim 68, further comprising:

a third receiver component for receiving a plurality of new ℓ-bit input plaintext blocks to replace a plurality of ℓ-bit plaintext blocks of said input plaintext string at index i; and

a seventh component for providing a data signing function that determines an authentication tag incrementally for each of the said plurality of new ℓ-bit input plaintext blocks.

70. An authentication system for providing a data signing function that determines an authentication tag, comprising:

a partitioner for partitioning the padded input plaintext string into a plurality of equal-size plaintext blocks of ℓ bits in length;

a randomization component for performing a randomization function over each of said plurality of plaintext blocks using a different

index for each plaintext block to create a plurality of input blocks each of ℓ bits in length;

a pseudo-random encipher component for applying a block cipher using a secret key over each of the said input blocks to create a plurality of enciphered blocks each of L bits in length;

a combining component for combining said plurality of enciphered blocks to create an authentication tag; and

further providing an out-of-order verification function for the authentication method comprising:

a receiver for receiving an input authentication tag for verification and a plurality of n plaintext blocks comprising ℓ bits each, each plaintext block being accompanied by a different index;

a randomization component for performing a randomization function over each of said plurality of plaintext blocks using said index for each plaintext block to create a plurality of input blocks each of ℓ bits in length;

a pseudo-random encipher component for applying a block cipher using a secret key over each of the said input blocks to create a plurality of enciphered blocks each of L bits in length;

a combining component for performing the same combination operation as that used at a signing method for determining an authentication tag over said plurality of enciphered blocks to compute an authentication tag;

a comparator for verifying integrity of the plaintext blocks by comparing the input authentication tag and the computed authentication tag.

71. A program product for providing a data signing function that determines an authentication tag for use in conjunction with transfer of

data using a communication channel or with data storage on storage media, comprising computer readable program code, including:

first code for partitioning said data into a plurality of data blocks;

second code which, for each of said data blocks, performs a randomization function over said data block to create an input block of the same size as that of said data block, said input block not including a block identifier;

third code for applying a pseudo-random function to each said input block to create a plurality of enciphered blocks; and

fourth code for combining said plurality of enciphered blocks to create an authentication tag.

72. The program product of claim 68, wherein the third code for applying the pseudo-random function applies a pseudo-random function that is a standard block cipher.

73. A program product for providing a data signing function that determines an authentication tag, comprising computer readable program code including:

code for partitioning an input plaintext string into a plurality of equal-size plaintext blocks of ℓ bits in length;

code for creating a random vector of ℓ bits in length;

code for performing a randomization function over said plurality of plaintext blocks and the random vector block to create a plurality of input blocks each of ℓ bits in length;

code for applying a block cipher using a secret key over each of said input blocks to create a plurality of enciphered blocks each of L bits in length; and

code for performing a combination operation over said plurality of enciphered blocks to create an authentication tag.

74. A program product for providing a data signing function that determines an authentication tag, comprising computer readable program code including:

first code for partitioning a padded input plaintext string into a plurality of n equal-size plaintext blocks of ℓ bits in length;

second code for performing a randomization function over said plurality of n plaintext blocks to create a plurality of input blocks each of ℓ bits in length;

third code for applying a block cipher using a secret key over each of the said input blocks to create a plurality of enciphered blocks each of L bits in length; and

code for performing a combination operation over said plurality of enciphered blocks to create an authentication tag.

75. A program product for an authentication method, which provides data integrity, comprising:

first code for receiving a string including a plaintext string and an input authentication tag for verification;

second code for partitioning said plaintext string into a plurality of n plaintext blocks comprising ℓ bits each;

third code for performing the same randomization function as that used at a signing method for determining an authentication tag over said plurality of plaintext blocks to create a plurality of input blocks each of ℓ bits in length;

fourth code for applying a block cipher using a secret key over each of the said input blocks to create a plurality of enciphered blocks each of L bits in length;

fifth code for performing the same combination operation as that used at a signing method for determining an authentication tag over said plurality of enciphered blocks to compute an authentication tag; and

sixth code for verifying integrity of the plaintext blocks by comparing the input authentication tag and the computed authentication tag.

76. The program product as defined in claim 72, further comprising:

seventh code for creating a secret random vector block of ℓ bits in length;

wherein the third code performs the same randomization function as that used at the signing method over said plurality of plaintext blocks and the secret random vector block to create a plurality of input blocks each of ℓ bits in length;

wherein the third code performing said randomization function further comprises:

code for deriving a random initial vector from said string presented for decryption;

code for generating a sequence of unpredictable elements each of ℓ-bit length from said random initial vector in the same manner as used at signing method; and

code for selecting n plaintext blocks from said string in the same order as that used at the signing method, and combining said selected plaintext blocks and the random vector with a different corresponding element of said sequence of unpredictable elements to obtain a plurality of input blocks, in the same manner as that used at the signing method.

77. The program product as defined in claim 72, wherein the third code for performing said randomization function further comprises:

code for using a secret, random initial vector shared between sender and receiver;

code for generating a sequence of unpredictable elements each of ℓ-bit length from said secret random initial vector in the same manner as used at signing method; and

code for selecting n plaintext blocks from said string in the same order as that used at the signing method, and combining said selected plaintext blocks with a different corresponding element of said sequence of unpredictable elements to obtain a plurality of input blocks, in the same manner as that used at the signing method.

78. A program product for providing a data signing function that updates an authentication tag incrementally, comprising:

first code for partitioning an input plaintext string into a plurality of equal-size plaintext blocks of ℓ bits in length;

second code for performing a randomization function over said plurality of plaintext blocks to create a plurality of input blocks each of ℓ bits in length;

third code for applying a block cipher using a secret key over each of the said input blocks to create a plurality of enciphered blocks each of L bits in length;

fourth code for performing a combination operation over said plurality of enciphered blocks to create an authentication tag, said combination operation having an inverse; and

further comprising:

fifth code for receiving an input plaintext string including a plaintext string and an input authentication tag;

sixth code for partitioning said plaintext string into a plurality of n plaintext blocks comprising ℓ bits each;

seventh code for receiving a new ℓ-bit input plaintext block to replace an ℓ-bit plaintext block of said input plaintext string at index i;

eighth code for performing the same randomization function as that used at a signing method, using index i, on said new input plaintext block to create a first input block and performing the same randomization function as that used at a signing method, using index i, on said plaintext block at index i to create a second input block, each of the said created input blocks having ℓ bits in length;

ninth code for applying a block cipher using a secret key to the first input block and the second input block to create a first enciphered block and a second enciphered block, each of L bits in length;

tenth code for performing the inverse of said combination operation used at a signing method for determining an authentication tag to the input authentication tag and said second enciphered block;

eleventh code for performing the said combination operation used at a signing method for determining an authentication tag to first enciphered block and the result of performing the inverse of said combination operation; and

twelfth code for outputting the result of performing said combination operation to the first enciphered block and the result of performing the inverse of said combination operation as the authentication tag.

79. The program product of claim 78, further comprising:

fourteenth code for receiving a plurality of new ℓ-bit input plaintext blocks to replace a plurality of ℓ-bit plaintext blocks of said input plaintext string at index i; and

fifteenth code for providing a data signing function that determines an authentication tag incrementally for each of the said plurality of new ℓ-bit input plaintext blocks.

80. An program product for providing a data signing function that determines an authentication tag, comprising:

first code for partitioning the padded input plaintext string into a plurality of equal-size plaintext blocks of ℓ bits in length;

second code for performing a randomization function over each of said plurality of plaintext blocks using a different index for each plaintext block to create a plurality of input blocks each of ℓ bits in length;

third code for applying a block cipher using a secret key over each of the said input blocks to create a plurality of enciphered blocks each of L bits in length;

fourth code for combining said plurality of enciphered blocks to create an authentication tag; and

further providing an out-of-order verification function for the authentication method comprising:

fifth code for receiving an input authentication tag for verification and a plurality of n plaintext blocks comprising ℓ bits each, each plaintext block being accompanied by a different index;

sixth code for performing a randomization function over each of said plurality of plaintext blocks using said index for each plaintext block to create a plurality of input blocks each of ℓ bits in length;

seventh code for applying a block cipher using a secret key over each of the said input blocks to create a plurality of enciphered blocks each of L bits in length;

eighth code for performing the same combination operation as that used at a signing method for determining an authentication tag over said plurality of enciphered blocks to compute an authentication tag;

ninth code for verifying integrity of the plaintext blocks by comparing the input authentication tag and the computed authentication tag.